IN THE CLAIMS

1. (Previously Presented) A method for managing memory in a multiprocessor system, comprising:

defining a plurality of processor coherence domains within a system coherence domain of a multiprocessor system, the processor coherence domains each including a plurality of processors and a processor memory;

providing shared access to data in the processor memory of each processor coherence domain only to elements of the multiprocessor system within the processor coherence domain;

providing non-shared access to data in the processor memory of each processor coherence domain to elements of the multiprocessor system within and outside of the processor coherence domain;

defining a plurality of partition coherence domains within the system coherence domain of the multiprocessor system, each partition coherence domain including at least one processor coherence domain, a router node, and a peer input/output node, each partition coherence domain providing memory coherence for its respective processor coherence domain, router node, and peer input/output node.

2. (Previously Presented) A method for managing memory in a multiprocessor system, comprising:

defining a plurality of processor coherence domains within a system coherence domain of a multiprocessor system, the processor coherence domains each including a plurality of processors and a processor memory;

providing shared access to data in the processor memory of each processor coherence domain only to elements of the multiprocessor system within the processor coherence domain; and

providing non-shared access to data in the processor memory of each processor coherence domain to elements of the multiprocessor system within and outside of the processor coherence domain;

providing a limited sharing vector for each processor memory, the limited sharing vector operable to identify only a portion of processors in the multiprocessor system.

3. (Original) The method of Claim 2, the limited sharing vectors each operable to identify processors only within their processor coherence domain.

4. (Previously Presented) The method of Claim 2, wherein each sharing vector comprises 32 bits and the multiprocessor system comprises more than 512 processors.

5. (Original) The method of Claim 1, the non-shared access comprising read-only access.

6. (Original) The method of Claim 1, the non-shared access comprising exclusive access.

7. (Original) The method of Claim 1, the non-shared access comprising transient access.

8. (Previously Presented) A method for managing memory in a multiprocessor system, comprising:

defining a plurality of processor coherence domains within a system coherence domain of a multiprocessor system, the processor coherence domains each including a plurality of processors and a processor memory;

providing shared access to data in the processor memory of each processor coherence domain only to elements of the multiprocessor system within the processor coherence domain; and

providing non-shared access to data in the processor memory of each processor coherence domain to elements of the multiprocessor system within and outside of the processor coherence domain;

determining whether an element requesting shared access to a processor memory is outside of the processor coherence domain of the processor memory; and

denying shared access if the element is outside of the processor coherence domain.

9. (Original) The method of Claim 8, wherein each processor and processor memory comprises an identifier having a set of most significant bits identifying the processor coherence domain of the element, further comprising determining whether the element requesting shared access is outside of the processor coherence domain based on the most significant bits of the element.

10. (Original) The method of Claim 9, further comprising determining whether the element requesting shared access is outside of the processor coherence domain of the processor memory by comparing the most significant bits of the identifier for the element to the most significant bits of the identifier for the processor memory.

11. (Previously Presented) A multiprocessor system for managing memory among a plurality of processors, comprising:

a system coherence domain;

plurality of processor coherence domains defined within the system coherence domain;

wherein the processor coherence domains each include a plurality of processors and a processor memory;

wherein the processor coherence domains are each operable to provide shared access to data in their processor memory only to elements of the multiprocessor system within the processor coherence domain and operable to provide non-shared access to data in their processor memory to elements within and outside of the processor coherence domain;

a plurality of partition coherence domains within the system coherence domain of the multiprocessor system, each partition coherence domain including at least one processor coherence domain, a router node, and a peer input/output node, each partition coherence domain providing memory coherence for its respective processor coherence domain, router node, and peer input/output node.

12. (Previously Presented) A multiprocessor system for managing memory among a plurality of processors, comprising:

a system coherence domain;

plurality of processor coherence domains defined within the system coherence domain;

wherein the processor coherence domains each include a plurality of processors and a processor memory; and

wherein the processor coherence domains are each operable to provide shared access to data in their processor memory only to elements of the multiprocessor system within the processor coherence domain and operable to provide non-shared access to data in their processor memory to elements within and outside of the processor coherence domain;

each processor memory further comprising:

a limited sharing vector for each data piece within the processor memory; and

the limited sharing vector operable to identify only a portion of processors in the multiprocessor system.

13. (Original) The multiprocessor system of Claim 12, the limited sharing vectors each operable to identify only processors within their processor coherence domain.

14. (Previously Presented) The multiprocessor system of Claim 12, wherein each sharing vector comprises 32 bits and the multiprocessor system comprises more than 512 processors.

15. (Previously Presented) The multiprocessor system of Claim 11, the non-shared access comprising read-only access.

16. (Original) The multiprocessor system of Claim 11, the non-shared access comprising exclusive access.

17. (Original) The multiprocessor system of Claim 11, the non-shared access comprising transient access.

18. (Previously Presented) A multiprocessor system for managing memory among a plurality of processors, comprising:

a system coherence domain;

plurality of processor coherence domains defined within the system coherence domain;

wherein the processor coherence domains each include a plurality of processors and a processor memory; and

wherein the processor coherence domains are each operable to provide shared access to data in their processor memory only to elements of the multiprocessor system within the processor coherence domain and operable to provide non-shared access to data in their processor memory to elements within and outside of the processor coherence domain;

each processor memory operable to determine whether an element requesting shared access is outside of the processor coherence domain of the processor memory and to deny shared access if the element is outside of the processor coherence domain.

19. (Original) The multiprocessor system of Claim 18, wherein each element comprises an identifier and the processing coherence domain of each element is defined by a set of most significant bits of the identifier for the element, the processor memory further operable to determine whether an element requesting shared access is outside of the processor coherence domain based on the most significant bits of the element.

20. (Previously Presented) The multiprocessor system of Claim 19, further comprising the processor memory operable to determine whether the element requesting shared access is outside of the processing coherence domain by comparing the most significant bits of the identifier for the processor memory to the most significant bits of the identifier for the element.

21. (Original) The multiprocessor system of Claim 11, the processor memory comprising a plurality of discrete memories.

22. (Previously Presented) A method for managing processor memory in a scalable processor system, comprising:

receiving a request from an element for shared access to data in a processor memory;

determining whether the element is outside of a processor coherence domain of the processor memory; and

denying the request for shared access if the element is outside of the processor coherence domain of the processor memory;

granting non-shared access to an element regardless of whether the element is within or outside of the processor coherence domain of the processor memory.

23. (Original) The method of Claim 22, further comprising:

including an identifier for the element in the request for shared access;

determining whether the element is outside of the processor coherence domain of the processor memory based on the identifier in the request.

24. (Original) The method of Claim 23, determining whether the element is outside of the processor coherence domain of the processor memory by comparing at least a portion of the identifier in the request to at least a portion of an identifier for the processor memory.

25. (Original) The method of Claim 24, wherein the portion comprises a set of most significant bits.

26. (Canceled).

27. (Previously Presented) A system for managing processor memory in a scalable processor system, comprising:

a computer processable medium; and

logic stored on the computer processable medium, the logic operable to receive a request from an element for shared access to data in a processor memory, determine whether the element is outside of a processor coherence domain of the processor memory, and deny the request for shared access if the element is outside of the processor coherence domain of the processor memory, the logic further operable to grant non-shared access to an element regardless of whether the element is within or outside of the processor coherence domain of the processor memory.

28. (Original) The method of Claim 27, wherein the request for shared access includes an identifier for the element, the logic further operable to determine whether the element is outside of the processor coherence domain of the processor memory based on the identifier in the request.

29. (Original) The system of Claim 28, the logic further operable to determine whether the element is outside of the processor coherence domain of the processor memory by comparing at least a portion of the identifier in the request to at least a portion of an identifier for the processor memory.

30. (Original) The system of Claim 29, wherein the portion comprises a set of most significant bits.

31. (Canceled).